Hateful Memes Identification with Multimodal Classifications

Ayush Singh
CSA, IISc
ayushsingh@iisc.ac.in

Preeti Sharma
EE, IISc
preetis@iisc.ac.in

Abstract

Hateful meme detection is not only fundamental in governing a healthy online environment, but it is also an important research topic that requires both visual and linguistic modeling skills and advanced knowledge of effectively combining multi-modal representations. In this paper, we utilized data from Meta's Hateful Meme Detection Challenge, which contains examples of memes that can be successfully classified only when their text and images are considered coherently.

We built three models, a baseline, a VisualBERT, and a CLIP fine tuning model. By leveraging large pre-trained models and fine-tuning, our best model, Model 3, achieves a 69 % accuracy. We presented our findings, analyzed and compared the results both quantitatively and qualitatively, and discussed potential future steps.

1 Introduction

Since the emergence of MySpace, social media has experienced tremendous growth and has reshaped our ways for communication. Any user can share, comment, and react to any other users, unlimited by physical distance. While individuals' opinions are more easily visible to others and can thus provide inspirations and entertainment, social media has also inevitably provided an expressway for hateful expressions to propagate, which in turn fuels provocations of anger, hatred, and extremism.

The effort to detect and thus strike down hate speech on social media is not new. Hateful conduct policies of social media platforms have constantly been maintained and updated. Besides, around 18,800 scholarly articles published with keywords of "hate speech detection" and "social media" since 2005. More recently, there has been great progress in using deep learning-based models, and more specifically transformer-based models, to detect hate speech. For example, AI-BERT+CNN received 0.9, 0.79, and 0.97 F1 scores on the Davidson, Founta, and Twitter Sentiment Analysis data respectively.

Despite the promising progress, we need to acknowledge that other forms of hateful conduct are also prevalent on social media and require different modeling. While most forms are unimodal, such as hate speech, video, and audio, a more unique form is memes, which use both image and texts. According to Iloh, memes are integrated in daily communication, serving as a symbolic reflection on cultures and communities. Moreover, they are an integral part in younger generations' communications, as 55 percent of 13- 35-year-olds share memes every week. Therefore, it is important to deter hate spreading through memes.

Because memes contain both images and texts, it has been incredibly hard to accurately detect and classify hate memes. The main challenge present for training a hatedetection model is for it to understand the coherence between the corresponding text and image in any given memes. As demonstrated in Fig. 1 from Kiela et al., the images and texts of memes often may appear innocuous

independently, but when taken together they have hateful implications. Furthermore, unlike typical visual-linguistic objects, memes can contain subtle visual cues that ultimately determines their polarity. In addition, memes are likely to contain references to specific and diverse contexts that are not commonly used, making learning more challenging. As a result, a multimodal approach that effectively bridges CV and NLP is necessary. Our project aims to build and train a model that, given an input - a meme (which contains both words and an image) can output a binary classifier that classifies the meme as either hateful or not. We utilize the data from Facebook's Hateful Meme Challenge. Our baseline model is multimodal with independently pre-trained text embeddings and visual embeddings (ResNet152). We also build a CLIP fine tune model and tried to build a better classifier.

Figure 1: Illustrative (not real) examples of multimodal hateful memes from Kiela et al. (2020). While the memes on the left column are hateful, the ones in the middle are non-hateful image confounders, and those on the right are non-hateful text confounders.

2 Related Work

Deep learning methods used to detect hate in memes can be roughly divided into three categories: unimodal models, multimodal models with unimodal pre-trainings, and multimodal models with mutlimodal pre-trains.

Under the unimodal category, the most notable are ResNet , Faster R-CNN , and BERT , with the former two as image classifiers and BERT as a transformer for language. The architecture of both ResNET and Faster R-CNN have been covered in class, and a detailed description of BERT is in the Method section below. In general, advantages of unimodal models is that there have been wellestablished research on these methods and implementations are thus straightforward. However, they are inadequate in our task of correctly classifying memes which contain two modalities. Therefore, given the time constraint, we decided to not implement any model from this category.

Multimodal methods with pre-trained image and text contain various models. Ones with simpler structures include models using simple fusion, where pre-trained text and image embeddings are concatenated. Other model have more complicated structures. For example, supervised multimodal bitransformers, which uses an image encoder that accept arbitrary number of inputs and outputs separate image embeddings. These image embeddings are then projected onto text token space with a learnable weight matrix. Models in this category include ViLBERT, VlBERT, and VisualBERT, which were all submitted around August 2019. Compared to the unimodal category, these models analyze both text and image representation of a meme, and thus are more capable of determining the coherence between text and image. Moreover, these models are independent of feature extractions and thus can be applicable to different tasks

Other methods use multimodal pre-training. Notably, the difference between this and the previous categories is mainly in the pre-training. Therefore, the backbone models remain largely unchanged. Models in this categories include Visual BERT trained on the COCO (Microsoft Common Objects in COntext) dataset that contains over 330K images suitable for image captioning , and ViLBERT

with Conceptual Captions pre-training, which contains images and image caption styles that are more diverse than the original COCO set. An advantage of these models is that the information of text and image is already integrated through the pre-training, and thus their relationship can be more accurately captured. However, while multimodal methods with unimodal pre-trainings can edit text or images encoders independently without having to retrain multimodally, the process for those with mutlimodal pretrainings is less efficient. Despite such shortcoming, models in this category have been shown to be more effective in classification.

While these three categories capture the three main approaches to our problem, other models on the Leaderboard for the Hateful Meme Challenge explored additional options. Zhu topped the leaderboard with an unseen AUROC of 84.50 and accuracy of 73.20 by using external labels such as race tags for face and head . Muennighoff was the 2nd place by ensembling five different multimodal models, achieving 83.10 and 69.50 AUROC and accuracy, respectively. While these methods are effective, our goal by completing this project is to gain learning experience of building multimodal models, and therefore, we focused on more understanding and re-implementing the multimodal models described in the previous two paragraphs.

3 Method

3.1 Multimodal Classification Overview

Multimodal classification has become an increasing important and popular area of research in the last decade. Combining data representations from different modalities, it provides a much more comprehensive analysis and understanding of current problems spanning many domains.

3.1.1 Data Fusion

A critical piece in working with multimodal data is how to combine drastically different representations into a coherent joint representation. For example, in Fig. 2, while the visual representations is dense and high-dimensional, the text representation is sparse. Therefore, an effective fusion algorithm is necessary to conquer this challenge. Fusion algorithm has evolved drastically during the past few years. Our paper will focus mainly the conventional fusion steps, which were used in both our baseline and VisualBERT models.

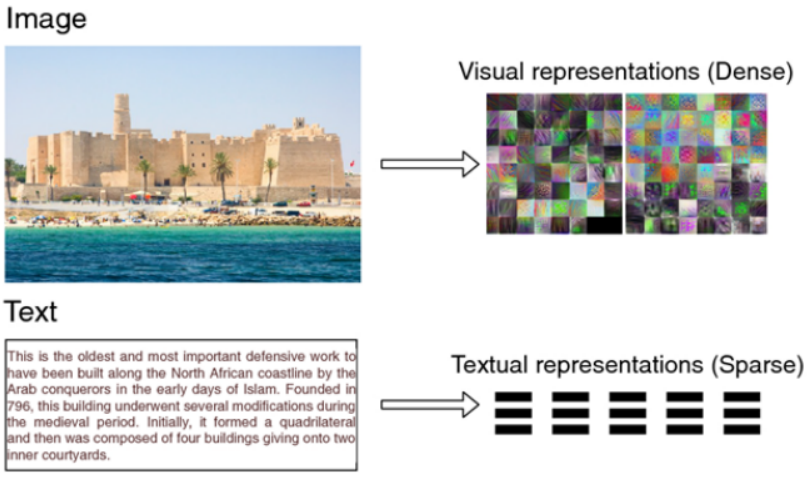


Figure 2: Challenge in combining different representations

Conventionally, the fusion step can occur in different stages of the classification process. Fusion methods can be categorized under either the model-agnostic approach, which are independent of the

specific models, or the modelbased approach [3]. In the model-agnostic approach, there are three commonly used approaches: early, late, and crossmodality fusion (see Fig. 3).

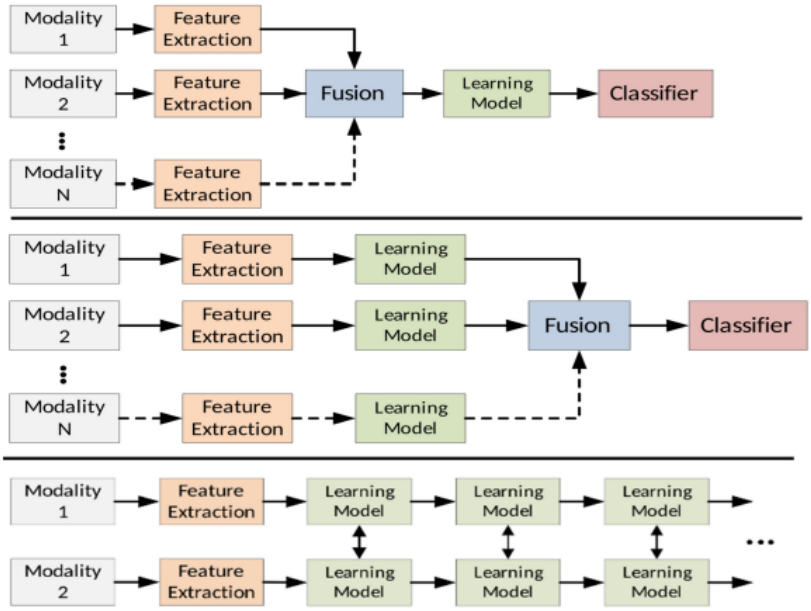


Figure 3: From top to bottom: early fusion, late fusion, crossmodality fusion

In early fusion, features of data are independently extracted from different modalities. This fusion occurs on the feature level, and thus learning occurs on the joined data. A disadvantage of this method is that fusing different modalities into one single representation can result in lack of homogeneity and thus larger prediction error.

In late fusion, each data source undergoes feature extraction and learning independently, followed by fusion at the decision-making stage. However, because of such independence, this method may fail to integrate data from different modalities well.

In Cross-modality fusion, however, fusion can occur during different stages of model training. Specifically, each modality can utilizes information from other modalities to improve performance. However, outputs from different streams may result in different shapes, thus increasing the difficulty of fusing.

3.1.2 Multimodal representation

Another challenge in multimodal classification which is also relevant for our project is how to represent the data for computational purposes. Baltrusaitis et al. introduced two categories of representations, joint and coordinated representations (Fig. 4). We will focus mainly on joint representation.

Concatenation is a basic joint representation method. Our baseline model uses feature concatenation, which puts features into one single vector (detailed below). Another category of methods use neural networks. For example, building on feature concatenation, Deep Concatenation combines features into a single deep learning layer. A more complicated method is Deep Merge, which is more commonly found in models with encoders. Each output node can take multiple input nodes across modalities. Lastly, Score Concatenation uses probability scores and such scores are combined with concatenation.

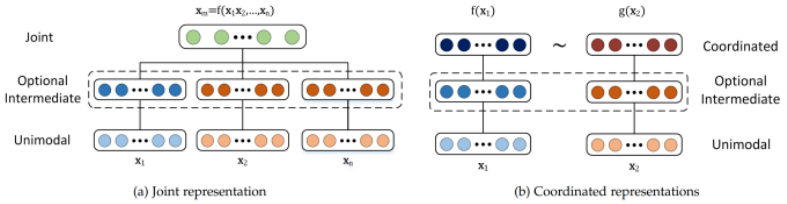


Figure 4: From top to bottom: early fusion, late fusion, crossmodality fusion

3.2 Baseline Model

The baseline model used two pre-trained models for text and vision that were fused and fine-tuned to our specific task of classifying hateful memes. The image data was converted to a format suitable for torchvision models and fed through the pre-trained Resnet152 model, outputting an image embedding in $\mathbb{R}^{B*1000}$. For the text data, we leveraged S-BERT's sentence embedding, which pools each word-level embedding of the sentence into a single sentence-level embedding. This outputs as a matrix in $\mathbb{R}^{B*768}$. The fusion method is a simple concatenation of the embedding dimension. Finally, the concatenated $\mathbb{R}^{B*1768}$ matrix is fed through a fully-connected linear and ReLU activation layer before projecting down to $\mathbb{R}^{B*1}$for the softmax scoring. We use a binary cross-entropy loss in order to backpropogate the errors.

3.3 VisualBERT Model

3.3.1 Overview

A model extremely relevant to our project is VisualBERT, which is a joint representation model for vision and language . VisualBERT is based on BERT, a Transformer that "pre-train[s] deep bidirectional representations from unlabeled text." VisualBERT reuses the self-attention mechanism from BERT, and building on BERT, it adds a set of visual embeddings for image modelling.

The visual embedding is a sum of three embeddings: the first embedding is a visual feature representation constructed using CNN; the second embedding serves as an indicator to differentiate a given visual embedding from a text embedding; the third embedding takes the aligned words and images' bounding regions as input and outputs the sum of positional embeddings corresponding to the words aligned with the image.

The visual embedding is then concatenated with the text embedding, as shown in Fig. 5 and fed into the transformer. This allows model to learn about the alignment between text and images and thus build a reasonable joint representation. The rest of the structure follows that of the BERT.

While traditional BERT involves only one pre-training process, VisualBERT splits that into task-agnostic and taskspecific pre-training. In our case, the task-agnostic pretraining is on COCO (introduced in Related Work section). Two procedures are done here: 1. the model is trained to predict some elements of masked inputs with vectors representing un-masked regions. 2. the model is trained to distinguish two captioning methods when more than one captions present for a given image; the first method contains two captions that corresponds to the image, while the second methods contains one corresponding caption but another randomly drawn caption. The task-specific training uses the target data with masked language modeling to familiarize the model with the specific task.

Followed by pre-training is fine-tuning, which closely resembles that of BERT. Provided with task-specific inputs and outputs, all the parameters are fine-tuned.

3.3.2 VisualBERT w/o Feature Extraction

The VisualBERT-based model for the Hateful Memes dataset relies on pre-trained VisualBERT weights, accessed using the Huggingface library. The inputs to the VisualBERT pre-training step

requires both visual embeddings and tokenized text. The text is fed through the BERT uncased tokenizer model, which is able to take into account out-of-vocabulary words to a certain extent by tokenizing based on word pieces. To make sure text inputs to the VisualBERT model are the same size, the BERT tokenizer pads the input up to a max length of 100 tokens with corresponding attention masks.

The visual embedding inputs for VisualBERT is fairly particular. Since the original model worked by aligning input text with regions within the image, its visual inputs are actually embeddings from an R-CNN. Because of this, the raw pixels are initially fed through a detectron2 RCNN model. The particular configuration of the detectron2 model uses a Resnet101 model as the backbone and boxes were chosen based on an non-maximum suppression score criterion. The R-CNN outputs a $\mathbb{R}^{B*100*1024}$ matrix which will be the visual embeddings for VisualBERT.

The visual and text inputs are used for the VisualBERT pre-trainer, which is based off the COCO dataset for a natural language and visual reasoning task. The last hidden state of this pre-trained model, which is in $\mathbb{R}^{B*200*768}$ is flattened and then fed through fully-connected feed forward layers with LeakyReLU activations. Layernorm and dropout regularization are also applied before and after the activations respectively. Finally, the matrices are projected $\mathbb{R}^{B*1}$ down to for softmax scoring in a binary cross-entropy loss function.

3.3.3 CLIP

In recent years, there has been a growing need for automated solutions for identifying and classifying hate speech and other types of offensive content on social media platforms. To address this challenge, researchers have proposed the use of deep learning models that can analyze both text and images in a multimodal setting. One such model is the CLIP (Contrastive Language-Image Pre-Training) architecture, which has emerged as a state-of-the-art approach for a wide range of tasks involving both text and images, such as image captioning, visual question answering, and object detection.

CLIP is a pre-trained neural network model that consists of an image encoder and a text encoder. The model is trained to predict which images were paired with which texts in a dataset, and in the process, it learns to extract meaningful features from both modalities. Compared to other deep learning models that are designed for a unimodal domain, the features extracted by CLIP are found to be more versatile and better suited for a multimodal domain.

To apply CLIP to the task of hate meme classification, researchers fine-tuned the model by freezing the backbone network up until the last second layer, and then added a classification layer as the final layer. The fully connected layer was trained with the help of Dropout Regularization and Leaky Relu activation functions. This fine-tuning process allowed the CLIP model to adapt to the specific task of identifying and classifying hate speech in memes, leading to improved accuracy and performance. Overall, the CLIP architecture is a promising approach for automated content moderation and could help address the growing problem of hate speech and offensive content on social media platforms.

4 Data

The Hateful Memes dataset was pre-processed by Facebook AI, as detailed in Kiela et al. After reconstructing the memes, the team hired annotators to filter the memes that were duplicates, not in English, violating, or containing slurs (which are unimodal). Then for each meme, five annotators rated it as "definitely hateful", "not sure", or "definitely not hateful". The memes with disagreements were removed and the resulting label for each meme is binary.

To increase the challenge, the team added benign confounders, which are created by minimally replacing the image or text in a meme so the originally hateful meme becomes non-hateful (see Fig. 1).

Therefore, within the hateful category, memes can be unimodal hate, where the text and/or images could already be hateful, or multimodal hate, where only the combination of the text and the visual

makes them hateful. Besides the non-hateful examples, the non-hateful category also contains benign confounders that, through either text or image replacement, and become non-hateful.

The final dataset we used contains 11,040 memes, with 8,500, 540, and 2000 memes used as train, dev, and test set respectively. All sets are claimed to be balanced with the following compositions: 10% unimodal hate, and 40% multimodal hate; 20% benign text confounder, 20% benign image confounder, and 10% non-hateful. That being said, all splits were found to be unbalanced with usually more negative samples (e.g., the train split contain 5450 non-hateful vs. 3050 hateful). Therefore, at training time, the train set is manually balanced by sampling the negative samples such that their count equals the number of positive samples.

More specifically, in the train split, the text has a mean and median of 12 and 10 words respectively. The images have an average height of 597.65, a minimum and maximum of height are 94 and 825, an average width of 523.60, a minimum and maximum of widths are 95 and 823. Most of the models that were tested relied only on the raw pixel and text data. Some examples are shown in Fig. 1

The image data was preprocessed by resizing to a height and width of 224 and normalizing across all images in the split. Examples are shown in Sec. 4. This pre-processing is necessary for both the baseline and VisualBERT-based models. For VisualBERT, the detectron R-CNN backbone takes in BGR pixel data instead of the usual RGB, so that was converted specifically for that model. There is some implicit feature extraction within the tokenization process and R-CNN backbone for the VisualBERT model.

5 Result

5.1 Experimental Details

Hyperparameters were chosen by validating on the development set provided. For the baseline model, the learning rate was chosen to be 1e-2 with the number of steps is 8000. A dropout regularization with probability 0.2 was included as well. For the VisualBERT models, the learning rate was 1e-5 and the number of steps is 8000. The VisualBERT models also used a dropout probability of 0.2. For both models, training batch size was maximized to the available GPU memory, which resulted in a batch size of of 32. We chose to use the Adam optimizer due to its robustness. Performance was evaluated by computing accuracy and the area under the receiver operating characteristic curve (AUROC), which were calculated using sk-learn's API . The AUROC is standard in terms of model comparison because it measures the because of its scale and classificationthreshold invariance. The accuracy, on the other hand, gives a more interpretable quantity of correct hits relative to the sample size.

5.2 Quantitative Results

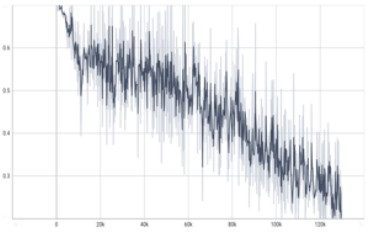
The results from these models show just how hard this problem of toxic meme classification really is. The VisualBERT-based model shows no improvements over the baseline simple concatentation fusion model despite being multi-modally pre-trained.

Model	Accuracy	AUC
Baseline	67.23 %	54.32 %
VisualBERT	65.48 %	52.34 %
CLIP	69.96 %	66.27 %

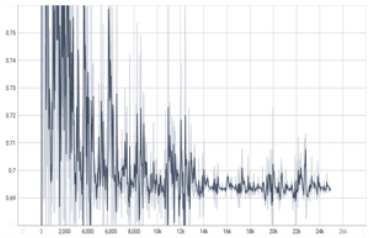
Table 1: Test Accuracy and AUROC for each model.

The models underfit the relationship between the input text/image data and the hatefulness of the meme, with the accuracy being barely better than randomly guessing. At the same time, increasing the number of training steps does not help the model converge either. This could be due to the relatively small number of training samples (6000). With the small sample size, the model is not able

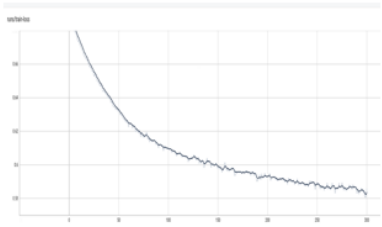
to fully learn the complex nature of memes such as the use of sarcasm. Alternatively, this underfit could also be explained by the model architecture and a feature space that is not rich enough.



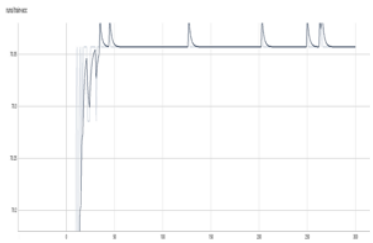
Training loss vs no. of steps for Baseline Model



Training loss vs no. of steps for VisualBERT Model



Training loss vs no. of epochs for CLIP



Accuracy vs no. of epochs for clip

5.3 Qualitative Results and Discussions

In this subsection, we will be analyzing mis-classified memes. Because of the nature of these memes, we want to make the reader aware of the hate-filled rhetoric before proceeding.

Trends that are consistent across three models are: (1). Each model classifies a given meme as Non-Hateful much more often . (2). With the hypothesis that a given meme is hateful, each model is more prone to have Type-I error (wrongly classifying a meme as non-hateful, bottom-left corner of a table) than Type-II error.

Trends that are specific to the model: (1) While VisualBERT w/o feature extraction (Model 2) has a better overall accuracy compared to the baseline model (Model 1), it correctly classifies hateful memes less successfully. To gain a better understanding of the trends discussed, we look into and divide misclassified images into some subcategories and provide examples in these sub-categories. Since we have 4^3 potential categories (each model has 4 categories) and even more when combining them, we decided to only look at certain important and major categories.

(1). Images that only Model 3 classified correctly. It is possible that some of these memes contain face with the correct racial tags. Specifically, the bottom two demonstrated how otherwise neutral texts paired with images with certain races are detected by Model 3 as hateful. (2). Images that Model 3 classified wrong but Model 2 classified correctly (see Fig. 9, 260 such cases). It is thus important to note the caveat of adding face tags here. For example, the top left may be wrongly classified as hateful because of its racial feature. Similarly, the racial tag is likely to be inaccurate with the lighting and number of people.

(3). Images that all models wrongly classified as nonhateful (see Fig. 10, 350 such cases). There are It seems that the models cannot recognize offensive images while the texts look neutral. One potential reason is that we don't have enough images during training for the model to learn that certain person's face (Hitler's, for example) often lead to hateful contents. Granted, it is incredible difficult for machine to learn about certain reference (Planet of Apes reference here and its relationship with Floyd, for example).

Figure 8. Examples that only Model 3 is correct. The top 2 are actually non-hateful, while the bottom 2 are actually hateful

Figure 9. Examples that only Model 2 is correct, but Model 3 is wrong. The top 2 are actually non-hateful, while the bottom 2 are actually hateful

Figure 10. Examples of Wrong Non-Hateful classifications

Figure 11. Examples of Wrong Hateful classifications

(4). Images that all models wrongly classified as hateful (see Fig. 11, 15 such cases). It seems that models deem many memes with offensive language (”killing” for example) and various profanity as hateful despite the meaning the memes are conveying are neutral or positive.

In conclusion, misclassifications above demonstrate that the models have difficulties understanding some images alone, and still have difficulties drawing meaningful connection between a given image and text. A potential solution for the former issue is to train on more image data, as our train set contains only 8500 images (6000 after balancing). Thus, some less frequent but offensive images can be successfully identified. A potential solution for the latter issue is to improve on the structure of our current best model.

The image and word embeddings are joined using a simple concatenation (early fusion). As discussed above, early fusion can lead to lack of homogeneity and larger prediction error. We can potentially try some more advanced fusion algorithm such as deep learning-based CNN or RNN fusions, as detailed in.

In addition, while external features can be incredibly helpful as shown by the large increase of accuracy of Model 3 compared to Model 2, we need to note that these tags can negatively bias the model and some tags could be wrong in first place. Therefore, further work is needed to improve the model architecture and feature space/fusion. One possible considerations for the feature space would be to leverage Web Entity Detection, as Zhu did in his winning solution in order to help the model understand real-world contextual information. Another possibility is to still use the race and gender tags but this time fuse this prior to the VisualBERT pre-training so that the information gets included in the VisualBERT representation.

6 Conclusion

In this paper, we worked on classifying hateful memes using three models: a baseline model that concatenates image embeddings from Resnet152 and sentence embeddings from S-BERT and feeds the concatenation through linear and ReLU layers; a VisualBERT model that concatenates image embeddings from R-CNN and text embeddings from BERT and feeds the concatenation through a Transformer. We observed that CLIP performed the best for their classification task, although

the dataset size was not sufficient for achieving high accuracy. On the other hand, VisualBERT's performance was slightly lower than the baseline model's, which raised questions about its suitability for this particular task. We identified several potential reasons for this, such as lack of relevant training data, domain mismatch, limitations of the model architecture, and noise in the data. However, we also noted that if the classification dataset is relatively simple and does not require complex reasoning or understanding of visual context, all three models may perform similarly, and the baseline model may be sufficient. Overfitting, similar architectures, and dataset bias are other factors that could explain their similar performance. Overall, we concluded that CLIP performed better than the other models for their classification task, but further experiments with more data and regularization techniques may be needed to improve generalization and achieve higher accuracy. As discussed above, the external features helped improve the accuracy, or, more specifically, the model's ability to correctly identify non-hateful memes at some cost of hateful meme classification accuracy. Given this result, if provided with more resource, we might consider adding other external features to further improve the accuracy and update the decision rule in our model to balance the false positive and false negative more.

References

1.ASEY FITZPATRICK. How to build a multimodal deep learning model to detect hateful memes. https:// www . drivendata . co / blog / hateful - memes - benchmark/, June 2020

2.Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. arXiv:1512.03385 [cs.CV], December 2015.

3.Constance Iloh. Do it for the culture: The case for memes in qualitative research. International Journal of Qualitative Methods, January 2021

4. Douwe Kiela, Suvrat Bhooshan, Hamed Firooz, Ethan Perez, and Davide Testuggine. Supervised multimodal bitransformers for classifying images and text. arXiv:1909.02950 [cs.CL], September 2019

5. Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. Microsoft coco: Common objects in context. ´ arXiv:10.48550/ARXIV.1405.0312, May 2014

6. Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github. com/facebookresearch/detectron2, 2019

7. Rong Zhu. Enhance multimodal transformer with external label and in-domain pretrain: Hateful meme challenge winning solution. arXiv:2012.08290v1 [cs.CL], 2020

8. https://paperswithcode.com/paper/hate-clipper-multimodal-hateful-meme

9. https://medium.com/codex/hateful-meme-detection-3c5a47097a08

10. https://towardsdatascience.com/how-to-get-high-score-using-mmbt-and-clip-in-hateful-memes-competition-90bfa65cb117